**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**

**Gráfico de barras

Descripción generada automáticamente con confianza media**

**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Caracterización de series temporales de tráfico de red mediante la transformada Wavelet**

**Autor: Álvaro Delgado Sancho**

**Tutor: Daniel Perdices Burrero**

**Junio 2024**

### Agradecimientos

Me gustaría mostrar mi agradecimiento, en primer lugar, a mi tutor Daniel Perdices por su gran ayuda y disposición a lo largo de todo el desarrollo de este trabajo, así como a la totalidad del departamento de HPCN (High Processing Computing and Networking) de la Escuela Politécnica Superior (EPS) de la Universidad Autónoma de Madrid (UAM).

Además, dar las gracias a todo el equipo docente de la EPS por acompañarme durante estos 4 años de grado poniendo su tiempo a disposición de una formación de calidad, y por su alta exigencia y motivación que me han ayudado a ser un mejor estudiante y profesional.

Por otro lado, agradecer a mis padres, Francisco Javier y María de los Milagros, y hermanos, Francisco Javier y Verónica, junto al resto de familiares y amigos por el afecto y apoyo incondicional mostrado durante todo este proceso de aprendizaje universitario. Finalmente, agradecer a mi pareja Paula por su cariño y motivación.

En resumen, agradecer a todas las personas que hayan podido ayudar o tener un impacto, con mayor o menor intensidad, en la realización de este trabajo fin de grado y que han hecho posible que este trabajo haya llegado a buen término.

### Resumen

El panorama actual de los sistemas de monitorización e identificación de intrusiones en redes de comunicaciones augura un futuro lleno de novedades y cambios en el paradigma de actuación. Ahora mismo, la mayoría de las organizaciones utilizan sistemas basados en firmas, los cuales poseen una serie de reglas que se actualizan periódicamente para detectar tráfico anómalo.

El objetivo de este trabajo es introducir un nuevo método de caracterización de series temporales en segmentos de red, de forma que la tarea de detectar comportamiento anómalo se facilite. La propuesta de este método se basa en la eliminación de la componente temporal en las series de cara a poder identificar series temporales con cierta similitud y poder así detectar comportamientos anómalos que no sigan un patrón relacionado con el tiempo.

A partir de esta idea se han desarrollado distintos métodos para conseguir esta idea, junto a una serie de pruebas y experimentos sobre un conjunto de datos masivo que nos han permitido observar el comportamiento de los algoritmos implementados.

Una idea a futuro acerca de este trabajo sería la de desarrollar métodos lo suficientemente potentes como para poder realizar este tipo de detecciones de manera automatizada descargando de responsabilidad y carga de trabajo a los administradores de sistemas y sus homólogos.

**Palabras clave**

Ondícula, clúster, centroide, series temporales, fragmento de red, clustering.

### Abstract

The current landscape of monitoring and network communication intrusion detection systems anticipates a future full of changes and novelties in the operational paradigm. Currently, most organizations use signature-based systems, which possess sets of rules that are periodically updated to detect anomalous traffic.

The objective of this work is to introduce a new characterization method for time series in network segments, thereby facilitating the task of detecting anomalous behavior. The proposed method is based on the removal of the temporal component from the series in order to identify time series with certain similarities and thus detect anomalous behaviors that do not follow a time-related pattern.

Based on this idea, multiple methods have been developed to achieve this idea, along with a series of tests and experiments on a massive dataset that have allowed us to observe the behavior of the implemented algorithms.

A future idea regarding this work would be the development of powerful enough methods to perform such detections automatically, thereby reducing the responsibility and workload of system administrators.

**Keywords**

Wavelet, cluster, centroid, time series, network segment.

### Índice

[Agradecimientos III](#_Toc168396860)

[Resumen 5](#_Toc168396861)

[Abstract VI](#_Toc168396862)

[Índice VIII](#_Toc168396863)

[Índice de figuras XI](#_Toc168396864)

[Índice de tablas XI](#_Toc168396865)

[Índice de cuadros XI](#_Toc168396866)

[1 Introducción 1](#_Toc168396867)

[1.1. Motivación 2](#_Toc168396868)

[1.2. Objetivos 2](#_Toc168396869)

[1.3. Estructura del documento 2](#_Toc168396870)

[2 Estado del Arte 3](#_Toc168396871)

[2.1. Motivación 3](#_Toc168396873)

[2.2. Objetivos 3](#_Toc168396874)

[3 Desarrollo 5](#_Toc168396875)

[3.1. Familiarización con los datos 5](#_Toc168396878)

[3.2. Representación gráfica 6](#_Toc168396879)

[3.3. Transformada Wavelet 8](#_Toc168396880)

[3.3.1. Definición teórica-matemática 8](#_Toc168396884)

[3.3.2. Aplicación 8](#_Toc168396885)

[3.4. Clustering 9](#_Toc168396886)

[3.4.1. Definición 9](#_Toc168396888)

[3.4.2. Aplicación 10](#_Toc168396889)

[4 Experimentos 13](#_Toc168396893)

[4.1. Subsección 13](#_Toc168396895)

[4.2. Subsección 13](#_Toc168396896)

[4.3. Subsección 13](#_Toc168396897)

[4.3.1. Subsubsección 13](#_Toc168396900)

[5 Conclusiones 15](#_Toc168396901)

[5.1. Subsección 15](#_Toc168396903)

[5.2. Subsección 15](#_Toc168396904)

[6 Trabajo Futuro 17](#_Toc168396905)

[6.1. Subsección 17](#_Toc168396907)

[6.2. Subsección 17](#_Toc168396908)

[Bibliografía 18](#_Toc168396909)

[Apéndices 20](#_Toc168396910)

[Apéndice A 21](#_Toc168396911)

[Apéndice B 23](#_Toc168396912)

### Índice de figuras

X.X Figura 1 XX

### Índice de tablas

X.X Tabla 1 XX

### Índice de cuadros

X.X Cuadro 1 XX

# Introducción

En los últimos años, investigadores y administradores de sistemas han tomado una gran relevancia en el ámbito de la monitorización de servicios de red con el objetivo de detectar comportamientos extraños y generar alertas para dichas situaciones. Esto sumado a la creciente capacidad y carga de servidores de empresas modernas y CPD (Centro de Procesado de Datos), ha hecho de este proceso algo bastante complejo y laborioso, debido a la variedad en el comportamiento de los servidores y subredes de organizaciones y grandes empresas. El objetivo principal de esta labor es la de encontrar soluciones que se adapten al comportamiento concreto de los servidores o segmentos de red a analizar. La gran cantidad de información y datos que poseen las series temporales, así como el limitado tiempo de los investigadores y administradores, dificulta su manejo y nos sugiere poner la vista en soluciones basadas en aprendizaje automático que puedan hacer frente al procesado masivo de datos.

La monitorización de segmentos de red es una herramienta muy útil para caracterizar las series temporales del tráfico de redes y poder así agruparlas respecto a un comportamiento similar. La agrupación de estos servicios nos permite ver un comportamiento común sobre el que poder detectar anomalías o situaciones de cambio relevantes en la dinámica del segmento de red.

La detección de estos cambios o anomalías suponen una gran ventaja para las organizaciones y empresas de cara a poder distribuir sus recursos de manera eficiente y lógica, y tener capacidad de previsión acerca de posibles subidas o bajadas de carga en sus servidores. La incapacidad para prever este tipo de situaciones podría suponer la pérdida de disponibilidad de servicios o el derroche de estos en una red infrautilizada, lo que se traduce en problemas económicos para la empresa.

Para la caracterización de estas series temporales se suelen utilizar métodos de clustering, que de manera automática agrupan los datos en el número de clústeres elegido teniendo en cuenta posibles relaciones entre los datos de su mismo clúster y diferencias más o menos significativas frente a datos de otros clústeres.

Para el proceso de agrupación de las series temporales, teniendo en cuenta la gran cantidad de información que poseen estas, sería necesario hacer uso de alguna técnica de reducción de componentes como PCA, t-SNE o UMAP. Posteriormente, se utilizarían métodos de clustering como k-means o DBSCAN para generar los resultados en base a la reducción de componentes previamente realizada.

Esta reducción de componentes nos facilita la visualización de los datos para su interpretación posterior, cuyo concepto surge de la idea de que la representación de una secuencia o grupo de datos puede basarse en un subconjunto más pequeño de variables respecto al inicial, manteniendo siempre medida de la varianza entre los datos. La varianza es la métrica que ayuda a distribuir los datos sobre los distintos clústeres y que mide la diferencia o variabilidad de los datos respecto al resto, por lo que al reducir el número de componentes se debe mantener dicha variabilidad para que la distribución de los datos se mantenga correctamente, que es justamente la idea que implementan los métodos de visualización como PCA (técnica que hemos utilizado).

Una vez están agrupados los datos en los distintos clústeres, la idea es poder categorizar las relaciones o patrones que puedan existir entre las series temporales pertenecientes a un mismo clúster, de manera que podamos encontrar información útil para poder utilizarla posteriormente en detección de anomalías o similares.

#### Motivación

Una manera inteligente de poder afrontar y detectar comportamientos anómalos es la de realizar predicciones, lo más precisas posibles, del comportamiento futuro de la carga de los segmentos de red. La longitud en el tiempo (horas, días, meses…) de estas predicciones puede variar dependiendo del tipo de comportamientos a detectar y del nivel de cambio en la carga de los segmentos de red.

Esta manera de actuar es eficaz en sistemas donde la carga del segmento de red está bastante relacionada con la componente temporal, es decir, segmentos de red en los que la carga durante un periodo temporal se repite de manera periódica y estable. Por lo contrario, en sistemas donde la carga es muy variable y fugaz, la predicción de anomalías futuras sería mucho más compleja, lo cual abre otro posible campo de investigación para mitigar este tipo de problemática.

Sin embargo, queremos encontrar una forma de eliminar la temporalidad en el proceso de caracterización de las series temporales para el posterior intento de detección de anomalías en dichas series. El hecho de eliminar el tiempo de los datos nos da la posibilidad de dejar a un lado los comportamientos periódicos y estacionales que puedan entorpecer la visualización de patrones de mayor relevancia. Esta idea puede suponer avances en la caracterización de series temporales, las cuales son aplicables a distintos campos de dentro y fuera de la informática.

#### Objetivos

El principal objetivo de este trabajo es el de investigar y encontrar nuevas estrategias de caracterización de series temporales. De manera más concreta, queremos investigar acerca de una estrategia de caracterización de series temporales basada en la eliminación de la componente temporal de las señales para poder encontrar patrones o similitudes que no tengan relación con la temporalidad de la señal.

El hecho de desarrollar un método de estas características que funcione de manera eficiente supondría un gran avance en materia de detección y caracterización de anomalías en los sistemas. Este tipo de anomalías no están completamente identificadas actualmente debido a la dificultad de detectarlas, por lo que esta investigación podría suponer un ámbito de estudio futuro.

Para todo ello, se han de elegir herramientas adecuadas, probadas con experimentación, que nos ayuden a desarrollar nuestro objetivo. Uno de los objetivos es hacer uso de la transformada Wavelet para poder llevar a cabo la idea previamente comentada.

Otro de los objetivos es poder llevar a cabo la representación gráfica de las series temporales en crudo, y en base al método desarrollado, poder realizar una posterior comparación de las similitudes y diferencias entre series temporales que se hayan clasificado de manera similar.

#### Estructura del documento

Inicialmente se introduce el estado actual del campo de estudio que se trata en este trabajo, indicando hasta donde se ha llegado en términos de investigación. Posteriormente se detalla el desarrollo de la investigación y los procesos realizados para su llegada a término. Tras ello se explican los diferentes experimentos y pruebas realizados para la comprobación de la hipótesis inicial. Finalmente se explican los resultados obtenidos y se sacan una serie de conclusiones e ideas para posibles trabajos futuros en este campo.

# Estado del Arte

Estado del Arte



#### Artículos relacionados

Motivación

#### Métodos preliminares

Objetivos:

* Objetivo1
* Objetivo2

# Desarrollo

A lo largo de la realización del proyecto, se ha desarrollado la idea general de manera organizada y siguiendo unos pasos muy estructurados. La primera tarea al comenzar un proyecto de investigación es siempre la de familiarizarse con las herramientas de trabajo que se van a utilizar, la información y datos que debemos tener en cuenta y proponer un método de trabajo, y en este caso no ha sido diferente.



#### Familiarización con los datos

En lo relacionado con los datos e información con los que se ha trabajado, desde un inicio tenía conocimiento de que los datos a utilizar para realizar los distintos análisis y pruebas pertenecen a servidores de una empresa energética. La cantidad de datos proporcionados era bastante elevada, más concretamente corresponden a la carga de los servidores de dicha empresa a lo largo de varios años, por lo que se ha tenido que acotar la totalidad de los datos para poder trabajar con ellos de manera más flexible y manejable.

Es por ello por lo que decidimos centrarnos en los datos recogidos durante 6 meses, entre el 1 de enero de 2021 y el 30 de junio de ese mismo año. De esta manera, se pudo trabajar de manera más flexible y concreta sobre una muestra de datos más pequeña.

Además, se me proporcionó un clúster con los datos de la empresa cargados en el que he realizado la mayoría de las pruebas y ejecuciones de programas para el análisis de los datos.

Adentrándonos en la estructura de los datos, estos se disponen en archivos diarios con la carga de los servidores en cada uno de los días registrados. En cada uno de los archivos se encuentra una tabla con distintas columnas como número de conexiones, bytes enviados/recibidos por segundo, referencias de tiempo, etc. Cada fila de estas tablas contiene la información de las distintas conexiones realizadas a un servidor sobre diferentes direcciones IP.

#### Representación gráfica

Una vez estudiada la estructura y amplitud de los datos que se han utilizado para probar la hipótesis, la siguiente tarea que teníamos que realizar era una representación gráfica de distintas series temporales.

Para ello, y debido a la gran variedad de datos en las distintas tablas, se consideró oportuno agrupar los datos según la dirección IP a la que se conectaban, recordando que cada fila de las tablas representa una conexión a un servidor. De esta manera, logramos agrupar las conexiones a direcciones concretas y estudiar las más relevantes, es decir, en nuestro caso, realizamos una ordenación de los datos a partir del número de conexiones realizados a dichas direcciones IP.

Una vez obtenidos los datos más relevantes (direcciones IP con mayor número de conexiones), la idea era representarlos y poder observar diferencias entre la carga sobre las diferentes direcciones. Para evitar representaciones con información demasiado escasa que impidiera un posible análisis o estudio, se restringió el proceso de representación gráfica a aquellas direcciones que tenían al menos 1000 conexiones.

Excluyendo las direcciones mencionadas, el resto de ellas que cumplían con el número mínimo de conexiones se representaron gráficamente. En el eje horizontal se representaba la componente temporal, que es la que más tarde buscamos eliminar, y en el eje vertical se indicaban los bytes totales recibidos, a los que podríamos denominar como “carga”. Sin embargo, para poder realizar comparaciones entre direcciones con distinta carga, se le aplicó una normalización a este eje vertical dividiendo cada valor por el valor máximo de bytes recibidos. De esta manera, las gráficas generadas tenían todas ellas valores verticales comprendidos en el intervalo [0, 1], lo cual hace que podamos tener a una misma escala todas las representaciones gráficas independientemente del tamaño de unas u otras.

Además, como es habitual en situaciones reales y profesionales, existen periodos de carga mínimo o inexistente en ciertos servidores, ya sea por mantenimiento, por problemas de funcionamiento o simplemente por el ciclo de trabajo de la propia empresa. Es por ello que dichos periodos de falta de datos se han sustituido de manera automática por ceros para evitar distorsionar demasiado las representaciones visuales de las direcciones que tengan estos comportamientos.

Gráfico, Histograma

Descripción generada automáticamenteGráfico, Histograma

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamenteGráfico, Histograma

Descripción generada automáticamente

\*Sustituir por gráficas normalizadas con leyenda, tipografía, fecha por tref…

Las imágenes anteriores muestran varios ejemplos de las posibles representaciones de la carga sobre distintas direcciones IP, en las que se pueden diferenciar de manera clara un comportamiento variable en cada una de ellas. Existen direcciones con una carga mayor y más continuada como puede ser la de la dirección 0.0.0.0 que difieren mucho con otras como la 10.200.101.39 que posee una carga bastante baja durante todo el periodo temporal excepto en un momento muy concreto que recibe un pico de carga bastante más alto de lo habitual.

El comportamiento de estas series temporales es muy diverso y cambiante entre unas y otras, y son precisamente esas diferencias las que, tras la transformación de las series y su caracterización eliminando el eje temporal, queremos poder estudiar y catalogar.

#### Transformada Wavelet



##### Definición teórica-matemática

\*Definir qué es una transformada Wavelet

La transformada Wavelet es una herramienta matemática utilizada para analizar señales no estacionarias, que son aquellas que cambian su frecuencia a lo largo de un periodo de tiempo. Para realizar este proceso, se hace uso de una serie de funciones bases, las Wavelet.

La potencia de las Wavelet reside en su capacidad para retener información tanto del tiempo como de la frecuencia para poder así realizar análisis locales del comportamiento de la señal. Además, estas funciones base ayudan a descomponer las señales y obtener sus diferentes niveles de resolución para poder analizar comportamiento multiescalar.

“La Transformada Wavelet no es estrictamente un método estadístico de reconocimiento de patrones, sino que es un método de preprocesamiento que permite que los datos sean expresados más sucintamente.” (<https://www.bing.com/ck/a?!&&p=056aeab1add9ad58JmltdHM9MTcxNzU0NTYwMCZpZ3VpZD0wNWJhMmRjZi00YzMxLTZlZTAtMjVhZC0zOWM4NGQ2YTZmZDUmaW5zaWQ9NTI0NQ&ptn=3&ver=2&hsh=3&fclid=05ba2dcf-4c31-6ee0-25ad-39c84d6a6fd5&psq=transformada+wavelet&u=a1aHR0cHM6Ly93d3cudW0uZWR1LmFyL29qczIwMTkvaW5kZXgucGhwL1JVTS9hcnRpY2xlL3ZpZXcvMjIvMjQ&ntb=1>)

##### Aplicación

Una vez comprendida la idea general de la transformada Wavelet, empezamos a trabajar con la librería pywt de python, que proporciona una gran cantidad de funcionalidad para este tipo de transformadas.

Para realizar una transformada Wavelet con esta librería existen varios métodos, sin embargo, nosotros hemos elegido el método continuo mediante la función cwt() que necesita una serie de parámetros. Inicialmente, se pasa la señal que deseamos transformar, que en nuestro caso se corresponde al eje vertical de nuestras representaciones, es decir, a los bytes recibidos. También se debe indicar el tipo de transformada que se le aplica a la señal. Previamente se han mencionado varias familias de transformadas, y en mi caso he utilizado la gaussiana para realizar esta transformación. Finalmente, se le pasa una lista de escalas sobre las que itera la propia transformada para proporcionar distintos niveles de detalle.

La función mencionada devuelve 2 estructuras con valores diferentes. La primera es un array de 2 dimensiones donde cada se devuelven los valores de escala y tiempo de la señal original. La segunda estructura devuelta es un array con las frecuencias respectivas a cada una de las escalas utilizadas en las transformaciones.

El concepto de escala en la transformada Wavelet se relaciona con la capacidad de estirar o comprimir la señal en búsqueda de diferentes características. Utilizando escalas bajas podemos comprimir la señal y realizar un análisis más detallado de la señal. Por lo contrario, con escalas altas podemos estirar la señal y recoger características o patrones de índole general. Debemos tener en cuenta que la escala es inversamente proporcional a la frecuencia, por lo que en escalas pequeñas estamos estudiando frecuencias altas, y viceversa.

A su vez, el concepto de tiempo permite analizar el cambio de las frecuencias a lo largo de la señal. La transformada Wavelet se compara con una ventana temporal que se mueve a lo largo de la señal para analizar las partes de la señal en distintas resoluciones temporales.

Es esta segunda lista de frecuencias la que nos interesaba para poder formar los descriptores numéricos para la señal. Dado que para cada una de las direcciones se generaba una lista distinta de descriptores, la manera de agrupar estas listas en un solo valor ha sido realizando una suma total de la lista.

Como se ha comentado previamente, la primera estructura devuelta contiene los coeficientes para las distintas escalas y tiempos de la señal. El objetivo de esta transformada era obtener descriptores numéricos para las distintas direcciones, por lo que al tener una matriz de 2 dimensiones se complica la manipulación de estos datos. De cara a facilitar dicho manejo de los datos, se planteó realizar una suma del eje Y de dicha matriz para así obtener simplemente una lista de valores. En resumen, tras realizar la operación de la transformada Wavelet sobre los datos de todas las direcciones IP, se obtienen listas de valores numéricos, es decir, los descriptores deseados, que identifican a cada una de estas direcciones. Para su posterior estudio y análisis, estos descriptores se guardaron en ficheros de texto plano.

Todo este proceso de la transformación de las señales, aunque pudiera parecer que simplemente se ha realizado en una ocasión, se ha tenido que realizar en 3 ocasiones, esto es debido a los diferentes estudios que hemos querido realizar sobre los datos. Para realizar un estudio más completo, se ha realizado este proceso para 3 escalas diferentes siendo estas diaria, semanal y exponencial.

Debemos tomar en cuenta que cada unidad de escala corresponde aproximadamente a 5 minutos, por lo que dependiendo de la escala que deseamos estudiar, tenemos que calcular el equivalente a un día o una semana. En el caso de la exponencial, el concepto es algo diferente. La escala introducida va saltando por las potencias de 2 hasta la escala máxima introducida, que en este caso ha sido 1024 (2^10). Con esto se consigue utilizar los datos más relevantes sin tener que ir pasando por todas las escalas.

Una vez terminados dichos procesos de transformación era el momento de procesar los descriptores de todas las direcciones IP y caracterizar dichos descriptores para buscar similitudes y diferencias entre series temporales.

#### Clustering



##### Definición

El clustering consiste en una técnica o conjunto de procesos cuyo objetivo es el de dividir conjuntos de datos en distintos grupos. Es una herramienta muy utilizada en aprendizaje automático para la detección de patrones en conjuntos masivos de datos, los cuales serían difíciles de identificar de manera manual debido a su tamaño. Concretamente, el clustering es un algoritmo de aprendizaje automático no supervisado, el cual no dispone de un entrenamiento previo con conjuntos de prueba y que tiene el objetivo de crear una organización de los datos en grupos que compartan algún tipo de estructura o patrón.

Los distintos grupos en los que se dividen los datos se denominan clústeres, los cuales concentran conjuntos de datos que poseen una determinada similitud entre ellos. Para el cálculo de dicha similitud entre datos existen distintas técnicas que brindan resultados diferentes. Algunas de las técnicas más utilizadas para esto son k-means, clustering jerárquico o DBSCAN.

En el ámbito de estos algoritmos, los conceptos de similitud y distancia se disponen esenciales para la comprensión de su funcionamiento. En el caso de k-means, que ha sido el algoritmo mayormente utilizado en este trabajo, estos dos conceptos se relacionan íntimamente.

Para determinar el clúster al que pertenece un punto concreto se utiliza la distancia, generalmente la euclidiana, que calcula la distancia entre 2 puntos, en este caso, el centroide de un clúster y el punto pendiente de asignar. Tras el cálculo de las distancias con todos los centroides, se elige aquel clúster cuyo centroide tenga la menor distancia de entre todas las existentes y se asigna el punto a dicho clúster.

Inicialmente, se deben elegir las cualidades o propiedades sobre las que el algoritmo debe centrarse para discriminar los datos y el número de grupos o clústeres en los que se quiere dividir una muestra, aunque este número debería ir variando para encontrar la mejor separación de los datos posible. Una vez elegidas esas propiedades, entra en la conversación el concepto de centroide, que se define como el punto medio en el que la suma de las distancias al resto de puntos del mismo clúster sea mínima.

Al inicio del algoritmo estos centroides se pueden asignar de manera manual, ya que de manera iterativa se irán actualizando hasta dar con los centroides reales. Tras la asignación inicial de los centroides, se van añadiendo puntos del conjunto total de datos y clasificándolos al clúster que posea su centroide a menor distancia de dicho punto. Después de estas clasificaciones se procede a recalcular los centroides de todos los clústeres de manera que se cumpla la condición descrita previamente.

Estos pasos se repiten de manera iterativa hasta llegar al número límite de iteraciones o hasta llegar a un punto donde los centroides no cambien de manera significativa.

##### Aplicación

El proceso completo de clustering explicado previamente necesita de distintas herramientas para poder trabajar de manera eficiente. Para ello, se ha hecho uso de la librería sklearn de Python, la cual contiene herramientas como k-means o PCA, necesarias para el procesamiento y agrupación de los datos, así como para la reducción del número de dimensiones de estructuras, respectivamente.

El algoritmo de k-means necesita un conjunto de datos completo para realizar la separación en distintos clústeres, por lo que, al tener nosotros los descriptores de las direcciones IP separadas, un primer paso necesario sería generar una lista de descriptores que contenga la totalidad de estos. Sin embargo, al realizar esta unión de los datos en un solo conjunto, perdemos la capacidad de distinguir direcciones concretas. Debido a esto, la mejor manera de poder identificar puntos concretos dentro de los clústeres es creando un diccionario, donde la clave se corresponde con la dirección IP representada, y el valor sea su descriptor. De esta manera, una vez se hayan representado gráficamente los clústeres, mediante las coordenadas de los puntos que queremos estudiar podemos extraer sus descriptores que a su vez están asociados a una dirección IP en el diccionario.

Una vez creados, tanto la lista con los descriptores como el diccionario, conviene transformar de manera inteligente los datos para trabajar con ellos de la manera más eficiente posible. Dado que cada uno de los descriptores está compuesto por una lista de valores numéricos, eso supone que la lista completa de descriptores es realmente una lista de listas, la cual sería más fácil de tratar si la convertimos en una matriz donde cada descriptor se corresponda con una fila. Además, conviene eliminar los puntos donde no tenemos datos medidos y que están representados con NaN y reemplazarlos con ceros para poder tratar más cómodamente con ellos.

Ahora el siguiente paso es definir una serie de parámetros que nos ayuden a identificar las diferencias entre datos. Lo primero es fijar un número de clúster en los que repartir los distintos puntos, que en nuestro caso se ha decidido que sean 3. Este número se ha elegido tras diferentes pruebas, ya que la representación habitual de los datos se compone de un clúster central con gran densidad de puntos, y un clúster a cada lado de este con mayor dispersión de puntos en ellos. Aunque se ha comentado que se deberían fijar unos centroides iniciales para comenzar el algoritmo, al crear un objeto KMeans ya se generan unos centroides por defecto, por lo que no es necesario introducir nuevos.

Para facilitar la visualización de los datos y reducir el número de dimensiones, utilizamos el algoritmo PCA, al cual le indicamos el número de componentes al que queremos reducir los datos y los transforma. Una vez transformados los datos solo queda el proceso de representar gráficamente los clústeres y sus puntos de manera ordenada, y utilizando un color diferente para los puntos de cada clúster.

En la siguiente imagen podemos encontrar un ejemplo de clustering con escala exponencial utilizando las herramientas mencionadas (KMeans y PCA).

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Como ya se comentaba previamente, la elección del número de clústeres que se han definido viene dado por la distribución propia de los datos, ya que el clúster central (azul) posee una mayor densidad de puntos que los clústeres que tiene a sus 2 lados (verde y naranja) que tiene datos más dispersos, e incluso algunos puntos que distan mucho de cualquiera de los clústeres, denominados habitualmente como outliers. Este tipo de puntos serán uno de los principales motivos de estudio, por lo que tienen una gran importancia.

Además de este tipo de representación gráfica, se han probado otras diferentes que han mostrado resultados no tan útiles como las utilizadas. Sin embargo, estas pruebas nos han dado la posibilidad de generar ciertas conclusiones acerca de los datos.

Finalmente, y tras haber representado los datos visualmente, debemos tener la posibilidad de escoger una serie de puntos de la gráfica y observar las series temporales relacionadas con ellos. Por ello se ha implementado un método de búsqueda que, introduciendo unas coordenadas, podemos obtener la dirección IP de los N puntos más cercanos a dichas coordenadas, siendo N el número de puntos que queremos encontrar.



# Experimentos

Con el objetivo de contrastar la hipótesis general del proyecto, se han realizado una serie de pruebas para corroborar o desmentir dicha idea. Teniendo en cuenta que la interpretación de observar gráficas “similares” es algo subjetiva, aunque se utilicen métricas muy objetivas, no podemos definir unos resultados completamente precisos acerca de estos experimentos, aunque cabe decir que, como valoración general, los resultados han sido positivos.

La realización de estos experimentos se ha realizado a partir de la fase de clustering de los datos, debido a que es donde comienzan a poder observarse de manera gráfica y visual los datos, y ha supuesto una gran mayoría de las horas invertidas en este trabajo.



#### Transformada Wavelet

En esta primera fase de pruebas, hemos experimentado con distintas maneras de realizar las transformaciones. Por un lado, se han realizado estas transformaciones proporcionando diferentes escalas y tipos de Wavelet para poder estudiar diferentes opciones.

Como ya se ha comentado previamente, lo correspondiente a cada unidad de escala son aproximadamente 5 minutos en tiempo real, por lo que para las escalas lineales se han introducido 12\*24 unidades para la escala diaria y 12\*24\*7 para la semanal, de manera que cada una tome el valor correspondiente a ese tiempo real.

Dado que nosotros poseemos datos correspondientes a 6 meses enteros, se realizaron intentos para poder realizar una escala de ese tiempo concreto. Sin embargo, la escala diaria y sobre todo la escala semanal han sido muy costosas computacionalmente para realizar las transformaciones respectivamente, por lo que se descartó el realizarla con una escala mucho mayor. Para poner en contexto, el tiempo necesario para realizar la transformación con escala diario era aproximadamente de 2 horas, y en el caso de la semanal, de casi 10 horas. Si llevamos este aumento del tiempo a horas necesarias para una escala mensual o mayor, los tiempos necesarios serían demasiados grandes como para realizar estas transformaciones de manera dinámica. Recordemos que estas transformaciones no se realizan una única vez, sino que se han ido realizando pruebas con distintas escalas y familias de Wavelet para obtener los resultados más convenientes de cara a la investigación.

Una vez tratado el concepto de escala y sus límites computacionales, toca comentar las diferentes Wavelet base utilizadas en dichas pruebas. Existen diferentes familias de Wavelet que podemos utilizar como base para nuestras transformaciones, como pueden ser la de Daubechies, Mexican Hat, Gaussiana o la de Haar.

Dado que las transformaciones solo nos proporcionaban descriptores numéricos y de esta manera era difícil comparar la conveniencia de utilizar unas u otras, tras obtener los descriptores numéricos era necesario realizar clustering para poder observar los resultados de manera visual y realizar dicha comparación.

Dado que las familias mencionadas previamente son las más importantes y utilizadas, son aquellas con las que hemos realizado transformaciones experimentales. Inicialmente, se realizó la prueba con la Wavelet Daubechies en sus diferentes versiones, sin embargo, los resultados tras la fase de clustering eran demasiado dispersos para poder comparar series temporales, caso que sucedía también con la Wavelet Haar, ya que ambas son muy parecidas en cuanto a su estructura.

Las Wavelet Gaussiana y Mexican Hat tuvieron mucho mejores resultados, proporcionando clústeres con una representación gráfica muy útil de cara a su posterior caracterización. Dado que la Mexican Hat es una wavelet algo más peculiar que la gaussiana, nos decidimos por utilizar esta wavelet de cara al resto de experimentos de clustering y caracterización, aunque como se comenta, los resultados con la wavelet Mexican Hat habrían sido muy positivos igualmente.

De manera uniforme y global, las salidas de las transformaciones con diferentes escalas y familias de wavelet proporcionaban la misma estructura de salida, la cual corresponde a un descriptor numérico, el cual es en sí mismo una lista de valores.

#### Clustering

En la fase de agrupación de las series temporales se han tomado en cuenta distintos aspectos de cara a la realización de pruebas. Dichos aspectos por examinar pueden ser las diferentes escalas que se quieren probar (diaria, semanal y exponencial), así como los distintos métodos de clustering existentes para realizar este tipo de experimentos.

En la fase de clustering, debemos separar las herramientas utilizadas en 2 tipos: clustering y visualización. Las herramientas de clustering propias que hemos utilizado han sido k-means, la cual forma clústeres asignando los datos respecto a la distancia más corta a un centroide, y DBSCAN, la cual basa su estrategia de agrupación en la densidad de los datos.

#### Tipos de clustering

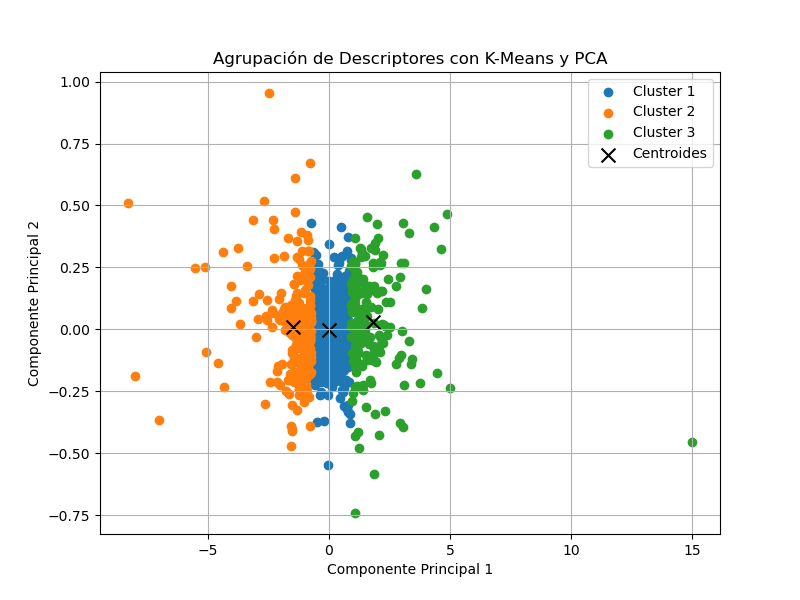


##### K-Means

Como se ha explicado previamente, el algoritmo k-means basa su funcionamiento en la asignación de puntos de datos en base al centroide que se encuentre más cerca. Este algoritmo es sencillo de utilizar y es bastante eficiente computacionalmente, sin embargo, es sensible al número de clústeres inicializados y a la asignación inicial de los centroides, que pueden otorgar resultados adversos dependiendo de los valores introducidos.

Otro punto a tener en cuenta es su forma, ya que se suelen obtener clústeres con distribución esférica, en los que pueden ser muy importantes sus puntos atípicos o outliers.

Simplemente, este método ha otorgado resultados muy buenos desde el principio, debido a su facilidad para implementarse y la capacidad de adaptación a los datos introducidos. En primer lugar, se intentó trabajar con un distinto número de clústeres para observar el distinto comportamiento del método. Dado que la forma habitual del conjunto de clústeres suele parecerse a una esfera, se querían recoger los datos más céntricos y diferenciarlos del resto, sin embargo, nos hemos encontrado con 3 grupos muy diferenciados: uno central, con mayor densidad de puntos; y otros 2, uno a cada lado del clúster central, con coordenadas negativas y positivas respectivamente. Teniendo en cuenta esta distribución de los datos, la asignación del número de clústeres se ha fijado en 3. Un ejemplo de esta distribución puede ser la siguiente imagen.



Otro aspecto a tener en cuenta es el método de visualización de los datos a utilizar. En este caso, se ha trabajado con PCA y con t-SNE, siendo ambos muy diferentes entre ellos. Mientras que PCA tiene como objetivo reducir la dimensionalidad intentando mantener el mayor valor de varianza de los datos posible, t-SNE hace lo mismo intentando mantener la similaridad entre datos.

Teniendo en cuenta que la anterior imagen está representada por PCA, este método ha mostrado una distribución de los datos bastante sencilla de interpretar y de trabajar con ella. En la imagen se pueden observar claramente los distintos clústeres, así como los outliers.

En cambio, la representación gráfica de los clústeres con el método t-SNE se muestra difícilmente interpretable. Esto se debe a su peculiar forma de “serpiente” que agrupa los datos de una manera poco intuitiva y en la que es difícil diferenciar entre puntos de diferente. Además, no queda muy clara la manera en la que diferencia las características de los 3 clústeres, ya que no se observan diferencias sustanciales entre los puntos de unos y de otros, como se puede ver en la siguiente imagen.

Gráfico

Descripción generada automáticamente

Como se comentaba anteriormente, la separación de los distintos clústeres que ha realizado el método de visualización t-SNE no deja muy clara la razón por la cual discrimina los puntos entre un clúster u otro, ya que existen puntos muy cercanos que pertenecen a otro clúster. La conclusión que hemos sacado de este método es que al tener que generar 3 clústeres exactamente, el algoritmo se ve obligado a mostrar esa separación, aunque no tenga una característica suficientemente discriminatoria para hacerlo.

Es por estas razones por las que hemos decidido trabajar con el método de visualización PCA, el cual es bastante amigable visualmente y nos ayuda a distinguir diferencias entre puntos.

##### DBSCAN

El método DBSCAN (Density-Based Spatial Clustering of Applications with Noise), como su nombre indica, basa su sistema de agrupación en la densidad de los datos. Como principal diferencia respecto a PCA, este algoritmo no necesita de una configuración inicial con el número de clústeres en los que se desea separar la muestra, sino que el propio algoritmo detecta automáticamente el número necesario. Esto supone una ventaja en cuanto a complejidad de uso, aunque se posee un menor control sobre los grupos en los que se quisiera dividir los datos.

Aunque este método es muy utilizado para realizar operaciones de clustering con múltiples objetivos, no nos ha proporcionado los resultados esperados de cara a poder comparar series temporales con comportamiento similar. La problemática surgida con este método es la aparición de los puntos demasiado dispersos a lo largo de la gráfica y la dificultad para poder observar la separación entre distintos clústeres. Otro motivo viene relacionado con este último y es el gran número de clústeres generados por el algoritmo en comparación con los 3 grupos en los que dividimos los datos con PCA, que en este caso ha sido de 10 clústeres diferentes. La división de los datos en tantos grupos diferentes complica claramente su estudio y comparación entre ellos, por lo que finalmente se ha optado por utilizar PCA como método principal de clustering para el resto de las pruebas.

Como se puede observar, no se muestran representaciones gráficas de los experimentos realizados con esta metodología. Esto se debe a la baja calidad de distribución y representación que nos ha proporcionado este método, por lo que se ha optado por no mostrar dichos resultados.

#### Escalas

Una vez probadas diferentes herramientas, tanto de clustering como de visualización, se ha optado para realizar el resto de los experimentos el método k-means para la partición de los datos y el método PCA para su visualización gráfica, que por supuesto conlleva la reducción del número de dimensiones de los datos. En esta sección vamos a ver los resultados obtenidos para las distintas escalas que hemos utilizado en la investigación.

##### Escala diaria

En este primer apartado, el objetivo es realizar un análisis detallado de la aplicación de las transformaciones y metodologías previamente explicadas sobre una escala diaria. Haciendo uso de escalas “pequeñas” como esta podemos detectar características con mayor detalle de las señales, es decir, comportamientos breves en el tiempo y que tengan el suficiente impacto.

En la siguiente imagen podemos observar la representación obtenida con los datos divididos en los 3 clústeres deseados.

Dado que la componente principal es la que posee una mayor cantidad de información a la hora de discriminar los datos, tiene mucho sentido que la división de los clústeres sea principalmente respecto al eje horizontal.Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Analizando la imagen nos encontramos con una zona céntrica muy poblada con forma redondeada donde encontramos la gran mayoría de los puntos. Aunque es la zona de mayor densidad y merece un análisis propio, también daremos mucha importancia a los puntos más alejado o atípicos de la gráfica, llamados outliers, ya que al ser más diferentes al resto nos pueden proporcionar un aporte de información diferente. Además, podemos ver la cercanía que tienen los centroides entre sí, lo que nos indica la gran densidad en torno a las coordenadas (0, 0) y la similitud entre los puntos de diferentes clústeres.

Gráfico, Histograma

Descripción generada automáticamenteGráfico, Histograma

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamenteGráfico, Histograma

Descripción generada automáticamente

Este grupo de imágenes representa las 4 series temporales más cercanas al punto (0, 0) de la representación de los clústeres, por lo tanto, pertenecientes al clúster central (color naranja). Como podemos ver, el nombre de cada una de las direcciones aparece como título de las gráficas.

A la hora de realizar un análisis sobre estas y futuras gráficas debemos tener en cuenta que el objetivo es buscar series temporales que tengan cierto parecido o similitud a simple vista y puede que, de manera desordenada. Esto es, la posible existencia de un pico el un momento muy inicial de una gráfica tiene parecido a otra serie que tenga ese mismo pico en un momento más avanzado. Esto se debe al método que estamos investigando, ya que la temporalidad queda al margen.

Observando las gráficas podemos observar que existen ciertos parecidos al igual que diferencias. Por ejemplo, tomando las 2 primeras gráficas, podemos adivinar la existencia de un comportamiento bastante estable y con poca carga media a lo largo de ambas señales. Tenemos además la existencia en ambas de varios picos puntuales de carga en distintos momentos del tiempo, por lo que este sería un ejemplo de lo que estamos buscando. La tercera gráfica tiene también comportamientos puntuales de mayor carga, sin embargo, posee saltos de unos puntos a otros debido a falta de información en esos momentos. Finalmente, tenemos una cuarta gráfica que difiera totalmente del resto, ya que posee una carga habitual mucho mayor y con carácter ascendente, lo cual no es el caso de las demás. Esta diferencia es lógica teniendo en cuenta la gran variedad de puntos que existe en una zona tan densa de puntos como es la central, unido a la forma en la que están calculados los descriptores, que es realizando la media entre el número de descriptores existentes. Este comportamiento es común, tanto al clúster central con centroide en las coordenadas centrales (0, 0), como para los otros 2 clústeres ubicados 1 a cada lado de este, por lo que se suponen las mismas conclusiones de los puntos más centrados de los 3 clústeres.

Cabe comentar que la obtención de las direcciones IP relativas a los puntos representados en la gráfica se realiza gracias al diccionario donde se guardan los descriptores de cada dirección, como se había comentado previamente. Una vez identificada la dirección IP podemos comparar su serie temporal con la del resto.

Aunque el estudio de la zona central es necesario e interesante, dado que existe una cierta dispersión de puntos alrededor del núcleo central, es importante realizar un análisis de los puntos más alejados de los centroides, que a su vez se encuentran relativamente cerca de otros puntos. El estudio de este tipo de puntos llamados outliers es importante de cara a identificar posibles series temporales anómalas o distintas al resto, cuya existencia es el propósito de este trabajo.

Tomando una pareja de puntos externos a la zona central podemos hacer un estudio de este tipo de outliers. Cogiendo siempre como referencia la representación gráfica del clúster, hemos querido comparar los puntos más cercanos a la zona inferior izquierda, más concretamente los puntos alrededor de la coordenada (-8, -0.35).

Gráfico, Histograma

Descripción generada automáticamente Gráfico, Histograma

Descripción generada automáticamente

Como podemos ver en este análisis, esto es un claro ejemplo del comportamiento que estamos buscando y tenemos como objetivo. Se ve claramente las diferencias entre ambas gráficas, sin embargo, vemos como se producen cambios en el comportamiento de las series en momentos temporales y situaciones diferentes. La principal característica es el cambio radical que presentan ambas series, pasando de unos niveles de carga bajos y estables hacia unos mucho más altas y fluctuantes.

Este tipo de cambio en el nivel de carga de un servidor es bastante notable, y el hecho de que ambos descriptores de las series temporales tengan similitud nos hace pensar en la posibilidad de detección de este tipo de cambios y variaciones.

##### Escala semanal

Al contrario que en la escala diaria, con esta escala semanal el objetivo es detectar características y patrones de largo plazo o comportamientos que afecten a la generalidad de la señal.

##### Escala exponencial

Este tipo de escala tiene cierta peculiaridad y difiere de las anteriores en la manera de trabajar en la transformada Wavelet. En las escalas anteriores introducíamos un límite de escala y la transformación se ocupaba de generar un análisis para cada una de las escalas entre 1 y el límite, sin embargo, en el caso de una escala exponencial se permite realizar un análisis más amplio. Dado que la escala va saltando de manera exponencial, en nuestro caso por las potencias de 2, podemos llegar a escalas mucho más altas a la vez que estudiamos un menor número de coeficientes. De esta manera, podemos captar de una manera más detallada patrones de comportamiento más detallados junto a algunos más generales. Además, mejoramos notablemente la complejidad computacional de estos métodos, ya que no se utilizan todas las escalas, sino tan solo las potencias de 2 hasta el límite superior, que en nuestro caso está fijado en 2^10. Finalmente, cabe comentar que este tipo de escala podría servir para detectar patrones de comportamientos logarítmicos que pasen desapercibidos con el uso de escalas lineales.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

# Conclusiones

El objetivo principal de este trabajo era realizar una investigación sobre un posible método de caracterización de series temporales basado en la eliminación de la temporalidad. Para efectuar todas las etapas de la investigación se han hecho uso de múltiples herramientas que nos han ayudado durante todo este proceso.



#### Objetivos cumplidos

En la fase inicial del trabajo se fijaron una serie de objetivos a cumplir a lo largo del proyecto, por lo que vamos a analizar cuales de ellos se han completado.

De manera general, el objetivo relacionado con el aprendizaje de todo el ciclo de vida de una investigación se ha cumplido con creces. Durante la duración total del proyecto se ha aprendido a tomar posibles soluciones a diferentes problemáticas, analizar la validez de los resultados obtenidos, así como la redacción de todos estos pasos para dejar reflejado todo el trabajo realizado.

En cuanto al tema técnico, pienso que se han obtenido resultados muy positivos e interesantes de cara a una investigación futura más completa. Es evidente que la capacidad de investigación en este trabajo es limitada y puede ser desarrollada mucho más a fondo para poder obtener resultados más valiosos, sin embargo, los resultados proporcionados nos hacen pensar en la posibilidad de grandes avances en el área de la caracterización de series temporales. Es por ello que creo firmemente que el propósito general del trabajo ha sido muy positivo y se han cumplido los objetivos previamente fijados de manera positiva.

#### Aprendizaje

#### A lo largo de todo el trabajo he aprendido el proceso completo del desarrollo de una investigación. Después de haber encontrado una idea sobre la que basar todo el trabajo, se ha trabajado sobre ella de manera organizada y completando una serie de pasos.

#### En el apartado más técnico, se ha trabajado con conceptos matemáticos, como la transformada Wavelet, así como con el análisis y comparación de conjuntos masivos de datos. Además, se han aprendido métodos de clustering y de visualización de datos previamente desconocidos, por lo que se han adquirido gran variedad de conocimientos de distinto ámbito, lo cual es bastante enriquecedor de cara al desarrollo personal y profesional.

#### Trabajo futuro

Tomando el trabajo de investigación realizado en este proyecto, sería interesante continuar de manera más profunda con dicha investigación y adentrarse en la capacidad de detección de anomalías a partir de metodologías como la desarrollada en este trabajo.

Como ejemplo, una de las mayores promesas actuales en el campo de la ciberseguridad es el desarrollo de IDS (Intrusion Detection System) basado en eventos y anomalías. En estos casos, podría ser interesante utilizar el método que hemos investigado acerca de la eliminación de la temporalidad de la carga de los servidores, o en otros posibles casos, del tráfico de red, para poder realizar avances en el campo de la detección de anomalías o ataques en sistemas.

Este trabajo está basado en la caracterización de las series temporales, sin embargo, no se ha profundizado en materia de detección de las anomalías. La manera de proceder ha sido la de realizar una caracterización de las series y una posterior búsqueda de similitudes entre las series. Esta búsqueda se ha realizado de manera visual y “humana”, por lo que automatizar esta tarea sería un posible campo de investigación hacia el futuro.

### Bibliografía

|  |  |
| --- | --- |
| [1] | [Qué es el clustering y cómo funciona: guía completa - MisApuntes (misapuntesdedatascience.es)](https://misapuntesdedatascience.es/que-se-entiende-por-clustering/) |
| [2] | [IEEE Xplore Full-Text PDF:](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9405648)  [3] [tnsm21.pdf (uam.es)](http://arantxa.ii.uam.es/~jlopezv/publicaciones/tnsm21.pdf)  [4] [Análisis de Componentes Principales (PCA) | Interactive Chaos](https://interactivechaos.com/es/wiki/analisis-de-componentes-principales-pca)  [5] [delga21/TFG\_UAM (github.com)](https://github.com/delga21/TFG_UAM/tree/main) |

### Apéndices

A

### Apéndice A

Este apéndice muestra el código utilizado para la representación gráfica de los datos separados por direcciones IP.

import pandas as pd

import matplotlib.pyplot as plt

import glob

# Time interval - 6 months

# procesa2-2021-01-01.csv - procesa2-2021-06-30.csv

files = glob.glob("shared-nvme/flujos\_cepsa/procesa2-2021-0\*.csv")

# Read CSV files

# Save it in a CSV or parquet??

dfs = []

for file in files:

    df = pd.read\_csv(file)

    dfs.append(df)

# DataFrame with all the data

combined\_df = pd.concat(dfs, ignore\_index=True)

# Plot every IP with more than 1000 points

for ip, filtered\_df in combined\_df.groupby("targetIP"):

    if filtered\_df.shape[0] < 1000:

        continue

    xx = filtered\_df.sort\_values("tref\_start", ascending=True)

    X = xx.tref\_start

    Y = xx.bpsPhyRcv / xx.bpsPhyRcv.max()

    plt.figure()

    plt.plot(X,Y)

    plt.title(ip)

    plt.savefig(f"shared-hdd/TFGAlvaro/plots/{ip}\_processed\_plot.png")  # Guardar el gráfico en la carpeta "plots" dentro de Jupyter

    plt.close()

B

### Apéndice B

Este apéndice muestra el código utilizado para la transformada Wavelet.

import pywt

import numpy as np

import matplotlib.pyplot as plt

import glob

import pandas as pd

files = glob.glob("shared-nvme/flujos\_cepsa/procesa2-2021-0\*.csv")

# Read CSV files

# Save it in a CSV or parquet??

dfs = []

for file in files:

    df = pd.read\_csv(file)

    dfs.append(df)

# DataFrame with all the data

combined\_df = pd.concat(dfs, ignore\_index=True)

wt='gaus1'

scaleMax = 12\*24 # 5min\*12=1h 1h\*24=1d

# Plot every IP with more than 1000 points

for ip, filtered\_df in combined\_df.groupby("targetIP"):

    if filtered\_df.shape[0] < 1000:

        continue

    xx = filtered\_df.sort\_values("tref\_start", ascending=True)

    X = xx.tref\_start

    Y = xx.bpsPhyRcv / xx.bpsPhyRcv.max()

    plt.figure()

    scales = np.arange(1,scaleMax,3)

    coef, freqs=pywt.cwt(Y, scales, wt)

    # Calculate descriptor adding the Y axis

    descriptor = coef.sum(axis=1)

    # Save the descriptor matrix in

    file = f"descriptors/day/descriptor\_{ip}\_exp.npy"

    with open(file, 'wb') as f:

        np.save(file, coef)